Masked Event Modeling: Self-Supervised Pretraining for Event Cameras

Simon Klenk[1]*, David Bonello[1]*, Lukas Koestler[1]*, and Daniel Cremers[1]
{simon.klenk, lukas.koestler, david.bonello, cremers}@tum.de
[1] Technical University of Munich * Equal contribution

Abstract

Event cameras offer the capacity to asynchronously capture brightness changes with low latency, high temporal resolution, and high dynamic range. Deploying deep learning methods for classification or other tasks to these sensors typically requires large labeled datasets. Since the amount of labeled event data is tiny compared to the bulk of labeled RGB imagery, the progress of event-based vision has remained limited. To reduce the dependency on labeled event data, we introduce Masked Event Modeling (MEM), a self-supervised pretraining framework for events. Our method pretrains a neural network on unlabeled events, which can originate from any event camera recording. Subsequently, the pretrained model is finetuned on a downstream task leading to an overall better performance while requiring fewer labels. Our method outperforms the state-of-the-art on N-ImageNet, N-Cars, and N-Caltech101, increasing the object classification accuracy on N-ImageNet by 7.96%. We demonstrate that Masked Event Modeling is superior to RGB-based pretraining on a real world dataset.

1. Introduction

Event cameras are promising imaging sensors for robotics and virtual reality applications. Event cameras contain independent pixels which trigger asynchronously once the observed brightness changes by a threshold [19]. They offer advantageous properties such as high temporal resolution, high dynamic range, low latency, and low power consumption. In recent years, the sensor's spatial resolution and signal-to-noise ratios have significantly improved, and researchers are increasingly using it in various applications. Due to the camera's different working principles, event cameras enable applications previously inaccessible for frame-based cameras, e.g. object classification in high-speed autonomous driving settings or eye and hand-tracking in low-power virtual reality systems.

The most successful frame-based approaches for high-level computer vision tap into large labeled training datasets. In fact, the remarkable progress in the field over the last decade can greatly be attributed to the availability of large labeled datasets, as well as improved network architectures with more parameters and increased compute power [58].

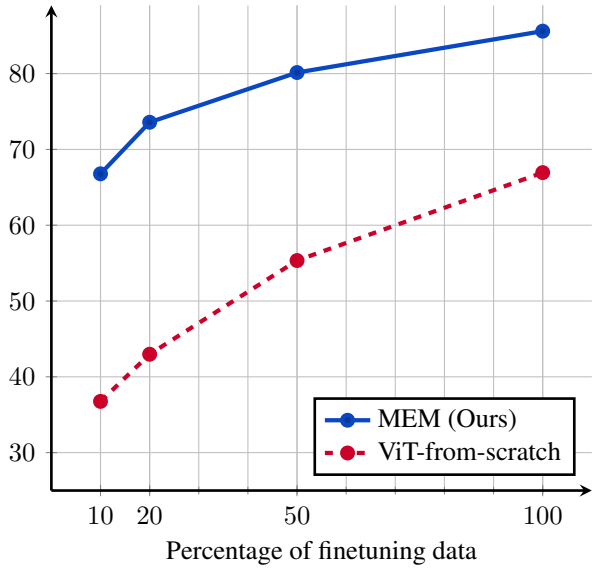

Figure 1. Classification accuracy on N-Caltech101 [44] with limited labeled data. We compare our self-supervised pretraining method (MEM) with a baseline. The original train set is split into subsets of decreasing size (100%, 50%, 20%, and 10%). *(blue)* MEM is pretrained on N-Caltech101 and uses the labeled subsets for finetuning. *(red)* Our baseline, ViT-from-scratch, uses the same architecture as MEM, however with a random weight initialization instead of pretrained weights. Both models have access to the same data and labels during finetuning. MEM consistently outperforms the ViT-from-scratch. The benefit of the proposed self-supervised pretraining becomes increasingly pronounced for tiny amounts of labeled data. The 10% subset only contains 650 samples for 101 classes.

While the event camera community can profit from many advances made in the frame-based domain, progress is still being held back by a lack of labeled event data [19, 55], as demonstrated by numerous approaches trying to solve this problem [7, 13, 27, 41, 45, 59, 62, 68, 72, 73]. Even with re-

cently released large-scale datasets like N-ImageNet [32], and N-EPIC-Kitchens [46], containing 1.3 million and 1 million event frames, all currently available event data still only makes up a tiny fraction of all vision data [41]. The need for larger datasets is amplified even more with the rise of the vision transformer (ViT) architecture [16] since it often requires significantly more training data to achieve superior performance over traditional CNNs [16]. However, ViTs are, in principle, well-suited for event data, as they do not make assumptions about a frame-like input structure but operate on general input sequences of tokens.

One solution to counteract the dependency on enormous labeled datasets is self-supervised pretraining. It has recently shown promising results in the NLP domain with BERT [14] as well as in the frame-based vision community [3,4,8,9,23,26,30]. Self-supervised pretraining divides training in two stages. In the first stage, a neural network is pretrained without labels solving a pretext task. For example, BERT performs pretraining on a large corpus of unlabeled text by masking several words and predicting them as the pretext task. In the second stage, the network is finetuned on a downstream task. Using pretraining can result in improved performance and often requires fewer epochs and labels than training a network from scratch (with random weight initialization). Fig. 1 visualizes the benefit of pretraining.

We want to leverage the methodology of pretraining in the event domain. Hence, we present Masked Event Modeling (MEM), a method that performs self-supervised pretraining on arbitrary event data recordings to alleviate the need for labeled event data at a large scale. Our approach is close to the recently proposed frame-based method BEIT [3] and inspired by the multitude of proposed extensions for other data modalities [2, 60, 63, 67]. After pretraining a ViT with our framework and finetuneing it on a downstream task, we consistently outperform all baselines from the literature.

The main contributions of this paper:

- We present the first framework for self-supervised pretraining on event data. During pretraining, our method does not require any labels or access to RGB image data, which makes it applicable to any event recording.
- We set a new state-of-the-art accuracy for image classification on all the datasets we used. Using our method, N-Imagenet surpassed the previous state-of-the-art by +7.96%, N-Cars +1.49%, and N-Caltech101 +9.5% (or +14% if using extra unlabeled event data).
- We show that the common practice of transferring RGB pretrained weights to the event domain is not always optimal. For example, on the N-Cars dataset, where the data originates from real-world recordings, we demonstrate that Masked Event Modeling achieves better performance than RGB-based pretraining, which relies on labels from ImageNet [12].
- The code and pretraining training checkpoints will be made available upon acceptance.

2. Related Work

This is the first work for self-supervised pretraining on event data. We thus present related work which performs self-supervised pretraining on frame-based data and methods that overcome the lack of annotated event data.

Self-supervised pretraining The idea of self-supervised pretraining is to first train a network on an unlabeled dataset by solving a pretext task [30] and then finetuning the network on a downstream task. The pretext task is defined such that the network learns basic visual features and intricacies of the data. After pretraining, the network is finetuned on a downstream task with a small labeled dataset in a supervised fashion. Examples of pretext tasks in vision are image reconstruction from masked or transformed input patches [1, 26], re-ordering of image patches [43], or predicting parameters of image rotations [22]. Early self-supervised approaches focused on CNNs as summarized in the survey by Jing et al. [30], whereas recently, a plethora of methods have been proposed for ViT architectures [3, 10, 26, 31].

One notable framework for ViTs is BEIT [3], which was largely inspired by the recent success of BERT [14] in the NLP domain. Like BERT, the pretext task of BEIT masks sections of the frame, intending to reconstruct these masked patches. However, instead of directly predicting the pixels of the high-dimensional masked patches, BEIT predicts visual tokens, which encode the semantic information of a patch in a single vector.

While BEIT and related methods [15, 18, 26, 61, 65, 71] essentially apply BERT-style pretraining onto images, several extensions to different data modalities have recently been proposed, e.g. Point-BERT [67] for point clouds, BEVT [63], VIMPAC [60] for video, and MultiMAE [2] for RGB images and depth or semantic maps. BEIT and its related methods largely inspire our method. However, in contrast to these methods, we investigate the proposed BERT-style pretraining strategy for the first time on event data. Ultimately, our method can be employed in domains where standard RGB cameras fail, e.g., in high dynamic range conditions or if high temporal resolution is required in a downstream task.

Overcoming the lack of labeled event data Although self-supervised pretraining has not been attempted for event cameras, numerous other works have proposed solutions against the lack of labeled events. Rebecq et al. [50] show

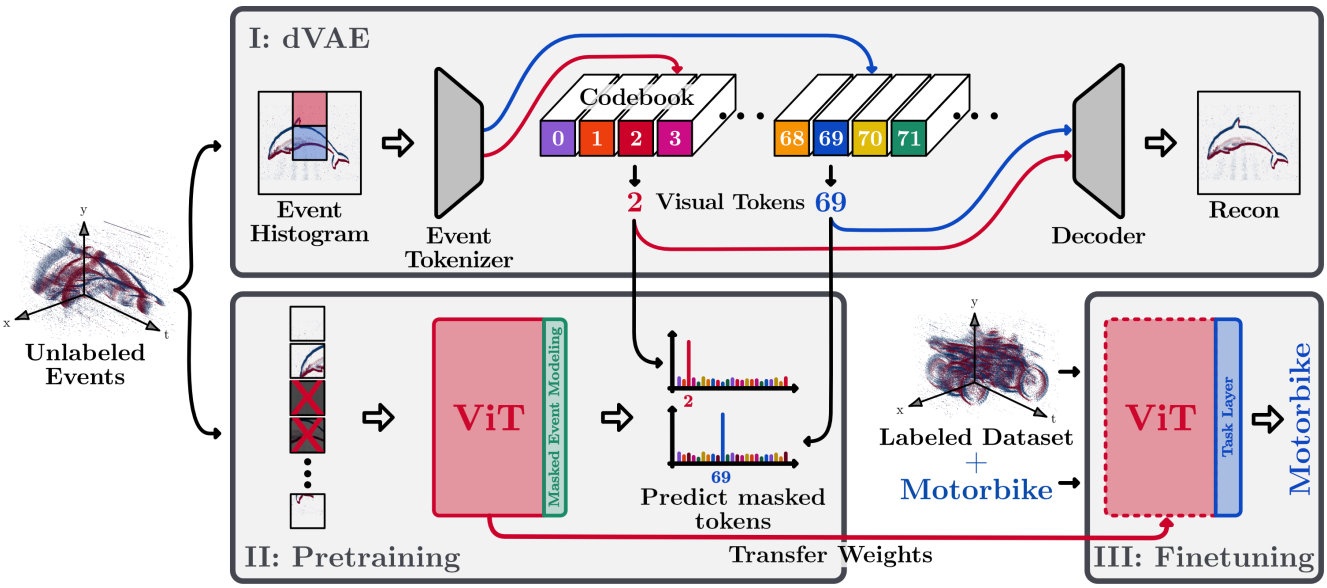

Figure 2. Overview of Masked Event Modeling (MEM). The proposed method consists of a three-stage pipeline. *(I)* In the first stage, we train a discrete variational autoencoder (dVAE) [47, 53] to compress the event histograms (input) to a list of discrete visual tokens. Each token – described by a fixed vector in the codebook – represents one input patch. The training objective in this stage is event histogram reconstruction. *(II)* In the second stage, we perform self-supervised pretraining of a ViT. The event histogram is divided into patches. We mask 50% of these patches, and the ViT predicts their corresponding (masked) visual tokens, similar to BERT [14], and BEIT [14]. The masked patches are replaced by a learnable embedding. Since the event tokenizer generates the ground truth, no labeling is needed. *(III)* In the final stage, the previously pretrained ViT can be finetuned on a downstream task. This is the only stage that requires labeled data.

that it is possible first to reconstruct grayscale frames from events and then perform object classification using standard frame-based networks. The drawback is increased computational demand and reconstruction artifacts, which can be severe for the event-to-image conversion.

Simulations can be used to leverage frame-based datasets by converting labeled frames to events. Model-based simulators such as ESIM [48] and v2e [28], supervised networks [20], or generative networks [72] can be used. The drawback of these methods is the large sim2real gap of the synthetic events. Furthermore, the video-to-event conversion only accurately simulates events with high framerate and current synthetic events still lack accurate noise characteristics [28].

Self-supervised training has been used to solve low-level event vision tasks. Zhu et al. [73] perform self-supervised learning of optical flow using events and images from a DAVIS camera [6]. Their approach is limited to estimating optical flow and requires a camera with access to time-synchronized and pixel-aligned grayscale frames and events. Parede-Vallés et al. [45] perform self-supervised intensity reconstruction using a generative event model assuming photometric constancy. In contrast to these works, we focus on the high-level object classification task, as this is a problem where traditional deep learning with large datasets can excel.

Zanardi et al. [68] use semi-supervised learning by starting from a supervised RGB teacher network which transfers its knowledge to an event-based student network. Similarly, Hu et al. [27] use the features of a pretrained RGB network as a backend and train a frontend network that translates event inputs to feature space. Both approaches [27, 69] require synchronous recordings of events and frames.

Recently, Wang et al. [62] and Messikommer et al. [41] proposed to leverage unpaired datasets of labeled frames and unlabeled events. Both approaches transfer knowledge from a powerful network in the RGB domain to the event domain. Sun et al. [59] proposed a similar approach designed explicitly for semantic segmentation. However, these methods depend on labels in the image domain and require both modalities' datasets to be captured in a similar scenario, which is not applicable in many applications.

In contrast to the above methods, we propose a general framework that performs self-supervised pretraining on event data, requiring no additional labels and no corresponding image data. In most applications, obtaining such unlabeled event data is very easy, whereas generating labels or accessing labeled image datasets from the same domain can be very costly and is often not possible in practice.

3. Masked Event Modeling

To overcome the lack of labeled event data, we adapt BERT-style pretraining to events. Masked Event Modeling closely follows the method by Bao et al. [3]. The events are preprocessed and then passed through the MEM pipeline, consisting of three main stages (see Fig. 2). The first two stages (dVAE and pretraining) operate on a potentially very large unlabeled event dataset. In the finetuning stage, we use the weights from the pretraining stage as initialization and train the neural network supervisedly on a target dataset.

3.1. Event Processing

The raw event data is preprocessed to an event histogram before entering the MEM pipeline, see Fig. 3 for examples. Since an event camera asynchronously reports brightness changes at a pixel, the sensor's output is a stream of individual events. Each event includes a polarity that indicates an increase or decrease in the observed brightness. For a moving camera and under constant illumination, brightness changes mainly occur at edges in the image plane. To obtain an image-like data structure with visible edges, we accumulate the events separated by polarities into a two-channel image $\mathbf{H} \in R^{(\mathrm{HxWx2})}$ using up to $N_{\mathrm{max\_evs}}$ events. We perform various data augmentations on this event histogram, which we detail in the supplementary material.

3.2. Discrete Variational Autoencoder

To reconstruct the masked event histograms during the pretraining phase, we first have to reduce the model's output space. Directly predicting raw histogram values of input patches would lead to a higher computational cost and can cause overfitting on low-level visual details [60]. Instead, we use visual tokens which summarize high-level semantic information in a single vector per patch. Additionally, the tokens are discrete and thus predicting a multi-modal distribution over tokens is straight forward.

To do this, we employ a discrete variational autoencoder (dVAE) following Ramesh et al. [47]. The essential idea is that each input patch of size $P_x \times P_y$ is compressed to a codebook vector $\mathbf{z} \in \mathbb{R}^d$, which summarizes the visual features of the patch. Each codebook vector has a unique index, called a visual token, which is later predicted by the ViT in the pretraining phase. Since the codebook is fixed during pretraining, the visual token (a single number) is sufficient to represent the semantic information of the patch.

The dVAE consists of three parts: the event tokenizer (encoder), the codebook bottleneck $\mathbf{Z} \in \mathbb{R}^{N \times d}$, and the decoder (refer to phase I in Fig. 2). The event tokenizer $q_\theta(\mathbf{z} \mid \mathbf{h})$ takes the full event histogram as input. Each input patch $\mathbf{h} \in \mathbb{R}^{P_x \times P_y \times 2}$ is mapped to a latent codebook vector $\mathbf{z}$. The decoder $p_\Phi(\mathbf{h} \mid \mathbf{z})$ learns to recover the event

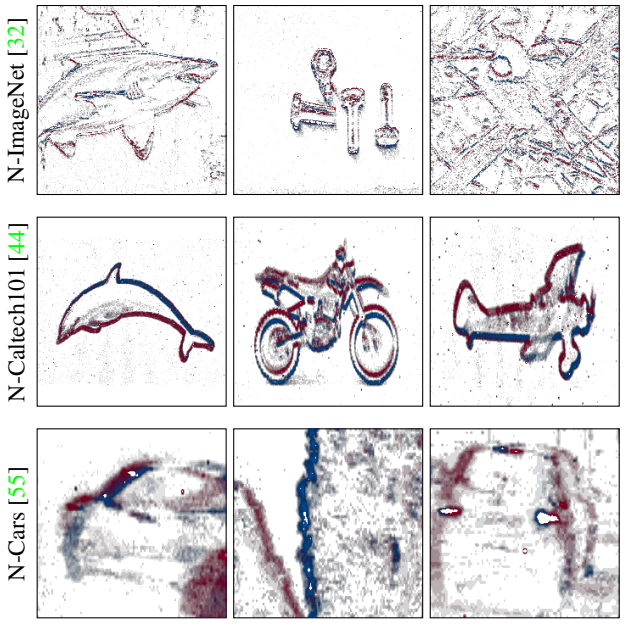

Figure 3. Example histograms from the datasets used in this work. In each row, we show three examples of N-Imagenet [32] (first row), N-Caltech101 [44] (second row) and N-Cars [55] (third row). Note that N-Cars is the only real event dataset and hence features different event statistics and noise distribution than the two other semi-synthetic dataset (refer Sec. 4 for details). Positive events are visualized in red, negative events in blue.

patch given the visual tokens. The training objective can be written as $\mathbb{E}_{\mathbf{z} \sim q_\theta(\mathbf{z}|\mathbf{h})}[\log(p_\Phi(\mathbf{h} \mid \mathbf{z}))]$. We place a uniform prior on the token distribution. This corresponds to maximizing the evidence lower bound (ELBO [34, 51]) of the log-likelihood of $p(\mathbf{h})$ [47]

$$\mathop{\mathbb{E}}_{\mathbf{z} \sim q_\theta(\mathbf{z}|\mathbf{h})}[\log(p_\Phi(\mathbf{h} \mid \mathbf{z}))] - D_{KL}[q_\theta(\mathbf{z} \mid \mathbf{h}), p_\Phi(\mathbf{z})]. \quad (1)$$

Due to the tokens being discrete, we employ the Gumbel softmax relaxation technique [29, 39] for obtaining the gradient (also used in related tasks [3, 47, 67]). We found that using gradient clipping often leads to more stable training when training dVAE on event histograms. We visualize the decoded tokens in Fig. 5, which shows that the tokens capture visual features such as lines and wedges.

3.3. Pretraining

The goal of pretraining is to train the backbone architecture (in our case, a ViT [16]) on self-supervised inputs for the network to gain an understanding of the data. First, the event histogram is divided into patches, and 50% of the patches are masked and replaced by a learnable mask embedding. The partially masked input is then used to train the pretraining network, which consists of the ViT and a tem-

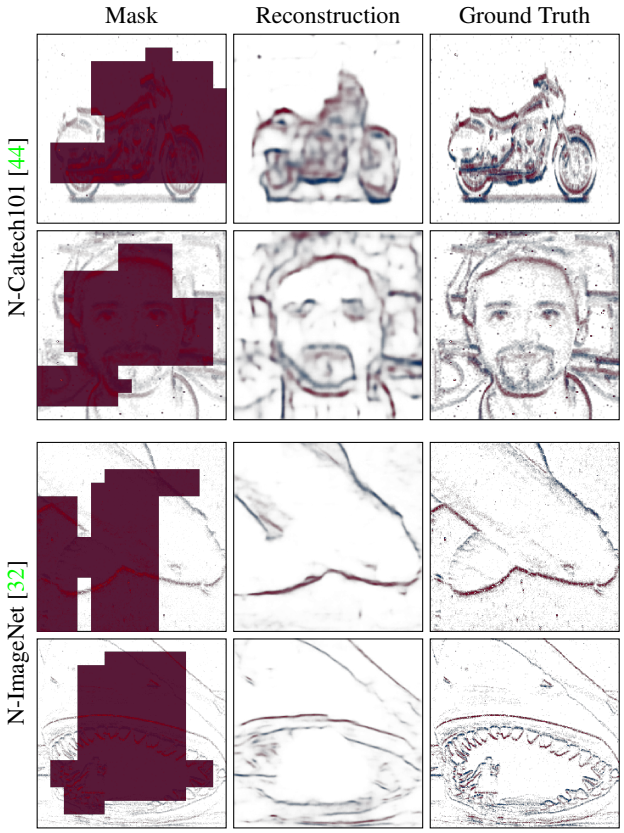

Figure 4. From left to right: We visualize the masked input histograms, the reconstructions during pretraining, and the ground truth, for N-Caltech101 [44] (top) and for N-Imagenet [32] (bottom). Note how the ground truth can be recovered even if large parts of the input to the ViT are masked. The ViT predicts the tokens for all patches, which are then decoded into the predicted event histogram by the decoder of the dVAE. Also note that these visualizations are rendered from the test set.

porary Masked Event Modeling (MEM) layer (inspired by [3], [14]). The MEM layer predicts the visual tokens $\mathbf{z}_{k^{\mathcal{M}}}$ of the masked patches $\mathbf{h}^{\mathcal{M}}$ during training. The training objective can be written as

$$\max \sum_{\mathbf{h}\in\mathcal{D}} \mathbb{E}_{\mathcal{M}} \Big[\sum_{k\in\mathcal{M}} \log(p_{\text{MEM}}(\mathbf{z}_k \mid \mathbf{h}^{\mathcal{M}})) \Big], \qquad (2)$$

where $\mathcal{D}$ is the dataset and p_{MEM} models the distribution of the visual tokens given the masked patches $\mathbf{h}^{\mathcal{M}}$. By inferring the tokens only from the non-masked patches, the ViT learns to model the semantic and spatial information of the event histograms. We visualize the predictions of the ViT on the masked event histogram in Fig. 4 on the test set .

3.4. Finetuning

In the last stage, we use the knowledge learned during pretraining to bootstrap the learning of the labeled dataset.

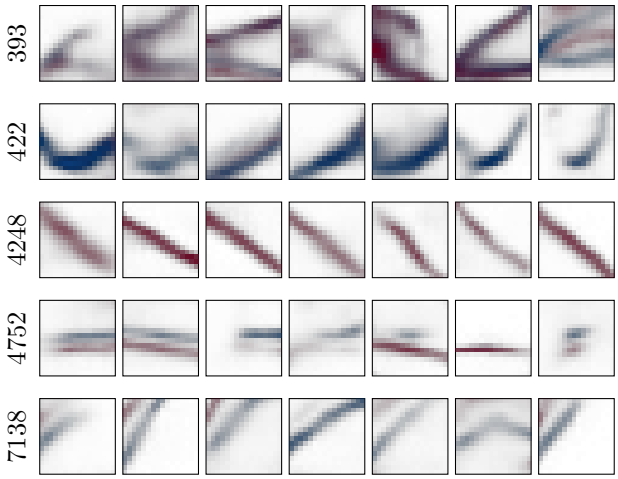

Figure 5. We visualize examples of decoded codebook vectors with the codebook index 393, 422, 4248, 4752 and 7138. Notice how each codebook index corresponds to a specific visual feature, e.g. a wedge of positive and negative polarity in the first row (393), or a diagonal red line in the third row (4428). Also note how the codebook employs the polarity information. The codebook size is 8092. These visualizations are rendered from the test set.

This is done by transferring the weights of the pretrained network to initialize the ViT. Only the final MEM layer is replaced by a task-specific layer. The dVAE is no longer used. Since the ViT has learned to complete a partially masked event histogram during pretraining, its weights can already effectively process event data for high-level vision tasks. Our method only requires labeled data when finetuning on a specific task.

4. Experiments

We evaluate the proposed Masked Event Modeling on object classification, a high-level vision task with established benchmarks and well defined performance measures. We report the top-1 classification accuracy on N-ImageNet [32], N-Caltech101 [44], and N-Cars [55]. We compare our method for each dataset (i) to multiple baselines from the literature and multiple of our baselines to investigate Masked Event Modeling in more detail. (ii) We compare MEM to training the same model with random weight initialization (ViT-from-scratch), and (iii) to supervisedly pretrained RGB-networks on ImageNet-1k and ImageNet-21k [12,52] (ViT-1k, ViT-21k). For MEM and all three proposed baseline methods (ViT-from-scratch, ViT-1k, and ViT-21k) we use the same implementation and hence share the data preprocessing. Please refer to the supplementary material for more details, such as comprehensive hyperparameters tables, train vs. test splits, and network architecture.

4.1. Object Classification

4.1.1 N-ImageNet

N-ImageNet [32] contains 1.78 million event streams recorded with a 480 × 640 Samsung DVS-Gen3 [56]. The event camera is moved in front of an LCD monitor, which displays images from ImageNet-1k [12]. Hence it contains 1000 classes. It is the largest event camera dataset for object classification. The second largest dataset ASL-DVS [5], is an order of magnitude smaller and features only 24 classes. N-ImageNet is a very challenging benchmark, as the best architecture so far has only achieved 48.94%, which is substantially below the 90% accuracy currently achieved on ImageNet-1k [66, 70].

In Tab. 1, we show that our proposed Masked Event Modeling outperforms the baseline method N-Imnet-EST [21] on N-ImageNet by +7.96%. Our baseline ViT-from-scratch can not reach on-par performance with the state-of-the-art because training a ViT on ImageNet is challenging due to the issue of overfitting [57] and the huge amount of computing required to reasonably sample the hyperparameter space on this large dataset. This demonstrates that employing MEM pretraining is an effective way to boost the performance for a very challenging event-based classification task, where a naive hyperparameter search might be too costly.

In the upper part of Tab. 1, which is colored in light gray, we show that our baseline implementations ViT-1k and ViT-21k achieve even better results. Note that the employed checkpoints for ViT-1k and ViT-21k perform pretraining on RGB ImageNet-1k and ImageNet-21k for 300 epochs (after an extensive hyperparameter search [57])[1]. Also note that N-ImageNet is the conversion of ImageNet-1k to the event modality[2]. Due to computational reasons, we only perform pretraining on N-ImageNet for 75 epochs. We believe longer training and hyperparameter optimization could further reduce the gap between RGB-based and event-only pretraining.

Method	Pretraining		Top-1
	Data	Labels	
ViT-1k	ImNet-1k	✓	61.90
ViT-21k	ImNet-21k	✓	65.00
N-Imnet-Hist [40]	✗	✗	47.73
N-Imnet-DiST [32]	✗	✗	48.43
N-Imnet-EST [21]	✗	✗	<u>48.93</u>
ViT-from-scratch	✗	✗	43.13
MEM (ours)	✗	✗	**57.89**

Table 1. Top-1 classification accuracies on N-ImageNet-1k [32]. MEM (ours) outperforms all baseline methods which only have access to event information. By using pretrained RGB-ImageNet checkpoints in the baseline methods ViT-1k and ViT-21, we show that an even higher accuracy can be achieved. All baseline numbers are taken from the N-ImageNet paper [32].

4.1.2 N-Caltech101

Similar to N-ImageNet, Neuromorphic-Caltech101 [44] features event streams recorded with an event camera moving in front of an LCD monitor while displaying images of the original RGB version Caltech101 [17]. It contains 8246 samples of 300 milliseconds duration, with 101 object classes. It has been a well-established benchmark dataset in the event literature, as demonstrated by the various baseline methods.

Method	Pretraining		Top-1
	Data	Labels	
HATS-Resnet [55]	ImNet-1k	✓	70.00
EST [21]	ImNet-1k	✓	81.70
DVS-ViT [64]	ImNet-21k	✓	83.00
E2VID [49]	ImNet-1k	✓	86.60
EventDrop [24]	ImNet-1k	✓	87.14
ACE-BET [36]	ImNet-1k	✓	89.95
ViT-1k	ImNet-1k	✓	92.06
ViT-21k	ImNet-21k	✓	91.10
HATS [55]	✗	✗	64.20
AEGNN [54]	✗	✗	66.80
AsynNet [42]	✗	✗	74.50
EvS-S [35]	✗	✗	76.10
ViT-from-scratch	✗	✗	66.94
MEM (ours)	✗	✗	<u>85.60</u>
MEM-NImNet (ours)	NImNet-1k	✗	**90.10**

Table 2. Top-1 classification accuracies on N-Caltech101 [44]. MEM (ours) outperforms all baseline methods which only have access to event information. However, MEM (ours) does not fully close the gap between event-only and supervised RGB-based pretraining methods. By pretraining our method self-supervisedly on N-ImageNet-1k (MEM-NImNet), it outperforms all baselines from the literature.

In Tab. 2, we show that our method outperforms all baselines which have access only to the labeled events of

[1] Furthermore, the baseline model ViT-21k has access to 14 million samples from ImageNet-21k [52].

[2] Note that the employed event datasets in this work solely contain grayscale events. Overall, RGB values are more information-rich than events because they contain (high-frequency) texture, color information, and detailed shading. Furthermore, events contain derivative-like information (changes in brightness) and hence carry much less information. Additionally, events are only triggered upon brightness changes above or below a certain threshold (usually 15-30% [19, 28]). Therefore, events are only a rough discretization of brightness value derivatives, which makes event-based object classification very challenging.

N-Caltech101 [44], setting a new state-of-the-art accuracy by +9.5% in this setting. While methods that have access to RGB-based pretraining do generally perform better than event-only methods on N-Caltech101 (upper part in light-gray vs. lower part of Tab. 2), our work significantly reduces this gap. If we grant our method access to the *unlabeled* event data of N-ImageNet in the pretraining stage, we show that this gap can be closed by setting a new state-of-the-art. We do this by pretraining MEM on the N-ImageNet [32] dataset for 75 epochs and finetuning MEM using the labeled event data from the N-Caltech101 training set. This method, called MEM-NImNet in the last row of Tab. 2, gives an additional performance boost of +4.5%. The fact that MEM can leverage event data from a different dataset demonstrates the generalization capability of our method. Note that MEM-NImNet only requires labels from N-Caltech101 and no labels from N-ImageNet, and solely has access to the event modality.

MEM-NImNet is only marginally outperformed by our baseline methods ViT-1k and ViT-21k. Our baseline ViT implementations perform slightly better than other methods using RGB ImageNet pretraining. We believe that this can be explained by the fact that we use modern data augmentations such as RandAugment [11], a cosine learning rate scheduler, and various regularization parameters which are inspired by state-of-the-art image-based ViT training schemes, in particular [57] and [40] (see supplementary material for details).

4.1.3 N-Cars

Both the N-ImageNet and the N-Caltech101 dataset are recorded by an event camera moving in front of a flat LCD monitor. The screen displays static RGB pictures of the original datasets under constant surrounding illumination. Due to this artificial way of recording, it is difficult to judge the performance of the evaluated methods on real event camera streams. Therefore we also benchmark MEM on the N-Cars dataset [55]. The N-Cars dataset contains 12,336 samples of the class car and 11,693 samples of the class background, where each sample is 100 milliseconds long. It was recorded by an event camera mounted behind the windshield of a car in an urban environment.

One fundamental difference between N-Cars and the other datasets is its different spatio-temporal event distributions since events are triggered by passing through a dynamic 3D scene. N-ImageNet and N-Caltech101 only capture event data originating from a homography [25] with respect to a flat LCD monitor. Furthermore, N-ImageNet and N-Caltech101 use constant light. In contrast, events of N-Cars are also triggered due to brightness changes in the outside world, e.g. by the blinking of a car's rear lights or traffic lights[3]. Such fluctuations frequently occur in real-world capturing conditions of N-Cars, as visualized in the last row of Fig. 3.

Method	**Pretraining**		**Top-1**
	Data	Labels	
HATS-Resnet [55]	ImNet-1k	✓	90.40
E2VID [49]	ImNet-1k	✓	91.00
EST [21]	ImNet-1k	✓	92.50
EventDrop [24]	ImNet-1k	✓	95.50
ACE-BET [36]	ImNet-1k	✓	97.06
ViT-1k	ImNet-1k	✓	98.00
ViT-21k	ImNet-21k	✓	96.24
HATS [55]	✗	✗	90.20
AsynNet [42]	✗	✗	94.40
EvS-S [35]	✗	✗	93.10
AEGNN [54]	✗	✗	<u>94.50</u>
ViT-from-scratch	✗	✗	92.71
MEM (ours)	✗	✗	**98.55**
MEM-NImNet (ours)	NImNet-1k	✗	93.27

Table 3. Top-1 classification accuracies on N-Cars [55]. MEM (ours) achieves a new state-of-the-art on this benchmark compared to all baselines. Since N-Cars contains real event data and only two classes, pretraining on ImageNet [12] or on N-ImageNet [32] does not perform as well as for this dataset. Our method requires much less data and compute since it only uses events from N-Cars.

In Tab. 3, we show that MEM pretrained only on N-Cars outperforms all presented baselines, including the methods which have access to RGB-based pretraining on ImageNet-1k and ImageNet-21k. Compared to ACE-BET [36], we raise the state-of-the-art by from 97.06% to 98.55%, an increase by 1.49%. While supervised RGB-based pretraining on ImageNet does boost performance on N-ImageNet (Tab. 1) and N-Caltech101 (Tab. 2), this effect is much weaker on the real-world dataset N-Cars, as can be seen by the similar performance of the upper and lower part of Tab. 3. We believe that the main reason for this is the different spatio-temporal event statistics of N-Cars compared to the semi-artificial N-ImageNet and N-Caltech101 datasets. This can further be supported by our experiment MEM-NImNet in the last row of Tab. 3, where the pretraining dataset is not N-Cars but N-ImageNet-1k. MEM-NImNet on N-Cars does not improve significantly over ViT-from-scratch, even though it has access to much more event data during pretraining. In contrast, pretraining on N-ImageNet achieves a remarkable finetuning performance on N-Caltech101 (see MEM-NImNet Tab. 2). This discrepancy clearly shows that performance improves

[3] An event camera captures these high-frequency oscillations of artificial lights due to its high temporal resolution.

Ablation	N-Caltech101	N-Cars
Full method	**85.60**	**98.55**
8x8 patches	80.81	96.13
32x32 patches	78.58	97.14
25% mask ratio	81.16	98.47
75% mask ratio	82.31	97.55
33% pretrain steps	81.17	95.16
No randAug [11]	80.67	98.14

Table 4. Ablation study of our method on N-Caltech101 [44] and N-Cars [55]. Default values for the full method are: patch size 16, masking ration 50% and using RandAugment [11]. The default image size is 224×224.

if pretraining and finetuning data characteristics are similar. Since our method only requires a few labels on the finetuning dataset (see Fig. 1) and does not rely on labels from a different modality (e.g. images), we enable effective pretraining by using only data of the specific target domain.

Furthermore, the experiments on N-Cars demonstrate that the common practice in the community of simply transferring RGB-pretrained weights to the event domain is not always the best option. This is especially the case if the target task entails specific event distributions which do not resemble RGB-based pretraining.

4.2. Masked Event Modeling with Few Labels

To demonstrate the usefulness of our method in applications where only very few labels are available, we train MEM with increasingly smaller subsets of the dataset. We split the original train set into mutually exclusive smaller subsets (100%, 50%, 20%, and 10%), where each smaller subset is contained in the larger one.

MEM is pretrained on the full N-Caltech101 [44] dataset *without labels*. Afterwards, we finetune the pretrained model on the smaller subsets. MEM is compared with the ViT-from-scratch baseline. Both models have access to the same data and labels during finetuning. In Fig. 1, we show that MEM consistently outperforms the baseline ViT-from-scratch. The benefit of MEM become increasingly pronounced for tiny amounts of labeled data. Although the 10% subset only contains 650 samples for 101 classes, MEM still achieves 66.78%, whereas ViT-from-scratch drops to 36.67% on this split.

4.3. Ablation Study

We study our method by changing single components and report the resulting top-1 accuracy on N-Cars [55] and N-Caltech101 [44]. All other hyperparameters are kept fixed during the ablation. We do not perform an ablation study on N-ImageNet [16] for computational reasons. In the first row of Tab. 4 we show the base accuracy of MEM using only data from N-Cars and N-Caltech101, respectively. We ablate the patch size of 16×16 by changing it to 8×8 and to 32×32, respectively. The patch size influences the number of visual tokens predicted by the dVAE and the number of patches used in the ViT. We find that the patch size of 16 x 16 yields the overall best result (on a default histogram size of 224×224).

Secondly, we ablate the masking ratio of 50% by changing it to 25% and to 75%. We find that the masking ratio of 50% is near the optimum on N-Caltech101 and N-Cars.

Furthermore, we ablate the length of MEM pretraining by only performing 33% of the pretraining steps. The significant drop in performance on both datasets shows that longer pretraining boosts MEM's performance. Finally, we ablate RandAugment [11] which results in a performance drop. Our empirical observations on N-ImageNet confirm that RandAugment is a very effective data augmentation for event histograms. This finding, together with the good performance of our baseline methods shows that using modern training techniques is highly beneficial for event data and can surpass event-specific methods.

5. Discussion

In a real application, obtaining large amounts of unlabeled data from a domain similar to the target domain is often quite easy, whereas labeling event data is very tedious. In fact, labeling event data is even more difficult than images because of the unconventional visual appearance of this data as demonstrated by the examples in Fig. 3 and its information deficit (see Sec. 4.1.1). One current limitations of our work is that it does not fully take advantage of the high temporal resolution of event cameras. We believe that novel real-world datasets that allow classifying objects at high speeds would first be required to evaluate such an extension of our method.

6. Conclusion

We introduce Masked Event Modeling, the first method for self-supervised pretraining on event data. MEM raises the state-of-the-art classification accuracies on the benchmarks N-ImageNet by +7.96%, N-Cars by +1.49%, and N-Caltech101 by +9.5% (compared with event-only methods), without requiring labels for pretraining.

Our work reveals that the common practice of simply using an RGB-pretrained network and applying it to an event vision task is not always optimal, especially if the finetuning dataset has different data characteristics. MEM can solve this problem by pretraining on the same type of data as the finetuning dataset. We believe that Masked Event Modeling will inspire future work to further unlock the potential of self-supervised pretraining and help to advance the field of event-based computer vision.

References

[1] Sara Atito, Muhammad Awais, and Josef Kittler. Sit: Self-supervised vision transformer. *arXiv preprint arXiv:2104.03602*, 2021. 2

[2] Roman Bachmann, David Mizrahi, Andrei Atanov, and Amir Zamir. Multimae: Multi-modal multi-task masked autoencoders. *arXiv preprint arXiv:2204.01678*, 2022. 2

[3] Hangbo Bao, Li Dong, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021. 2, 4, 5

[4] Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *arXiv preprint arXiv:2105.04906*, 2021. 2

[5] Yin Bi, Aaron Chadha, Alhabib Abbas, Eirina Bourtsoulatze, and Yiannis Andreopoulos. Graph-based object classification for neuromorphic vision sensing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 491–501, 2019. 6

[6] Christian Brandli, Raphael Berner, Minhao Yang, Shih-Chii Liu, and Tobi Delbruck. A 240× 180 130 db 3 μs latency global shutter spatiotemporal vision sensor. *IEEE Journal of Solid-State Circuits*, 49(10):2333–2341, 2014. 3

[7] Hu Cao, Guang Chen, Jiahao Xia, Genghang Zhuang, and Alois Knoll. Fusion-based feature attention gate component for vehicle detection based on event camera. *IEEE Sensors Journal*, 21(21):24540–24548, 2021. 1

[8] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9650–9660, 2021. 2

[9] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 2

[10] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9640–9649, 2021. 2

[11] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 702–703, 2020. 7, 8, 12

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 2, 5, 6, 7

[13] Yongjian Deng, Hao Chen, Huiying Chen, and Youfu Li. Learning from images: A distillation learning framework for event cameras. *IEEE Transactions on Image Processing*, 30:4919–4931, 2021. 1

[14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 2, 3, 5

[15] Xiaoyi Dong, Jianmin Bao, Ting Zhang, Dongdong Chen, Weiming Zhang, Lu Yuan, Dong Chen, Fang Wen, and Nenghai Yu. Peco: Perceptual codebook for bert pre-training of vision transformers. *arXiv preprint arXiv:2111.12710*, 2021. 2

[16] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2, 4, 8, 12

[17] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *2004 conference on computer vision and pattern recognition workshop*, pages 178–178. IEEE, 2004. 6

[18] Tsu-Jui Fu, Linjie Li, Zhe Gan, Kevin Lin, William Yang Wang, Lijuan Wang, and Zicheng Liu. Violet: End-to-end video-language transformers with masked visual-token modeling. *arXiv preprint arXiv:2111.12681*, 2021. 2

[19] Guillermo Gallego, T. Delbrück, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, Stefan Leutenegger, A. Davison, J. Conradt, Kostas Daniilidis, and D. Scaramuzza. Event-based vision: A survey. *IEEE TPAMI*, 2020. 1, 6

[20] Daniel Gehrig, Mathias Gehrig, Javier Hidalgo-Carrió, and Davide Scaramuzza. Video to events: Recycling video datasets for event cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3586–3595, 2020. 3

[21] Daniel Gehrig, Antonio Loquercio, Konstantinos G Derpanis, and Davide Scaramuzza. End-to-end learning of representations for asynchronous event-based data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5633–5643, 2019. 6, 7

[22] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018. 2

[23] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020. 2

[24] Fuqiang Gu, Weicong Sng, Xuke Hu, and Fangwen Yu. Eventdrop: Data augmentation for event-based learning. *arXiv preprint arXiv:2106.05836*, 2021. 6, 7

[25] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 7

[26] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022. 2

[27] Yuhuang Hu, Tobi Delbruck, and Shih-Chii Liu. Learning to exploit multiple vision modalities by using grafted networks. In *European Conference on Computer Vision*, pages 85–101. Springer, 2020. 1, 3

[28] Yuhuang Hu, Shih-Chii Liu, and Tobi Delbruck. v2e: From video frames to realistic dvs events. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1312–1321, 2021. 3, 6, 12

[29] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016. 4

[30] Longlong Jing and Yingli Tian. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 43(11):4037–4058, 2020. 2

[31] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *ACM computing surveys (CSUR)*, 54(10s):1–41, 2022. 2

[32] Junho Kim, Jaehyeok Bae, Gangin Park, Dongsu Zhang, and Young Min Kim. N-imagenet: Towards robust, fine-grained object recognition with event cameras. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2146–2156, 2021. 2, 4, 5, 6, 7, 12, 14, 15

[33] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 15

[34] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 4

[35] Yijin Li, Han Zhou, Bangbang Yang, Ye Zhang, Zhaopeng Cui, Hujun Bao, and Guofeng Zhang. Graph-based asynchronous event processing for rapid object recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 934–943, 2021. 6, 7

[36] Chang Liu, Xiaojuan Qi, Edmund Y Lam, and Ngai Wong. Fast classification and action recognition with event-based imaging. *IEEE Access*, 2022. 6, 7

[37] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 15

[38] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 15

[39] Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016. 4

[40] Ana I Maqueda, Antonio Loquercio, Guillermo Gallego, Narciso García, and Davide Scaramuzza. Event-based vision meets deep learning on steering prediction for self-driving cars. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5419–5427, 2018. 6, 7

[41] Nico Messikommer, Daniel Gehrig, Mathias Gehrig, and Davide Scaramuzza. Bridging the gap between events and frames through unsupervised domain adaptation. *IEEE Robotics and Automation Letters*, 7(2):3515–3522, 2022. 1, 2, 3

[42] Nico Messikommer, Daniel Gehrig, Antonio Loquercio, and Davide Scaramuzza. Event-based asynchronous sparse convolutional networks. In *European Conference on Computer Vision*, pages 415–431. Springer, 2020. 6, 7

[43] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*, pages 69–84. Springer, 2016. 2

[44] Garrick Orchard, Ajinkya Jayawant, Gregory K Cohen, and Nitish Thakor. Converting static image datasets to spiking neuromorphic datasets using saccades. *Frontiers in neuroscience*, 9:437, 2015. 1, 4, 5, 6, 7, 8, 12, 14, 15

[45] Federico Paredes-Vallés and Guido CHE de Croon. Back to event basics: Self-supervised learning of image reconstruction for event cameras via photometric constancy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3446–3455, 2021. 1, 3

[46] Chiara Plizzari, Mirco Planamente, Gabriele Goletto, Marco Cannici, Emanuele Gusso, Matteo Matteucci, and Barbara Caputo. E2 (go) motion: Motion augmented event stream for egocentric action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19935–19947, 2022. 2

[47] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021. 3, 4

[48] Henri Rebecq, Daniel Gehrig, and Davide Scaramuzza. Esim: an open event camera simulator. In *Conference on robot learning*, pages 969–982. PMLR, 2018. 3

[49] Henri Rebecq, René Ranftl, Vladlen Koltun, and Davide Scaramuzza. Events-to-video: Bringing modern computer vision to event cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3857–3866, 2019. 6, 7

[50] Henri Rebecq, René Ranftl, Vladlen Koltun, and Davide Scaramuzza. High speed and high dynamic range video with an event camera. *IEEE transactions on pattern analysis and machine intelligence*, 43(6):1964–1980, 2019. 2

[51] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*, pages 1278–1286. PMLR, 2014. 4

[52] Tal Ridnik, Emanuel Ben-Baruch, Asaf Noy, and Lihi Zelnik-Manor. Imagenet-21k pretraining for the masses. *arXiv preprint arXiv:2104.10972*, 2021. 5, 6

[53] Jason Tyler Rolfe. Discrete variational autoencoders. *arXiv preprint arXiv:1609.02200*, 2016. 3

[54] Simon Schaefer, Daniel Gehrig, and Davide Scaramuzza. Aegnn: Asynchronous event-based graph neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12371–12381, 2022. 6, 7

[55] Amos Sironi, Manuele Brambilla, Nicolas Bourdis, Xavier Lagorce, and Ryad Benosman. Hats: Histograms of averaged time surfaces for robust event-based object classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1731–1740, 2018. 1, 4, 5, 6, 7, 8, 12, 13, 14, 15

[56] Bongki Son, Yunjae Suh, Sungho Kim, Heejae Jung, Jun-Seok Kim, Changwoo Shin, Keunju Park, Kyoobin Lee, Jinman Park, Jooyeon Woo, et al. 4.1 a 640× 480 dynamic

vision sensor with a 9μm pixel and 300meps address-event representation. In *2017 IEEE International Solid-State Circuits Conference (ISSCC)*, pages 66–67. IEEE, 2017. 6

[57] Andreas Steiner, Alexander Kolesnikov, Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer. How to train your vit? data, augmentation, and regularization in vision transformers. *arXiv preprint arXiv:2106.10270*, 2021. 6, 7

[58] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pages 843–852, 2017. 1

[59] Zhaoning Sun*, Nico Messikommer*, Daniel Gehrig, and Davide Scaramuzza. Ess: Learning event-based semantic segmentation from still images. *European Conference on Computer Vision. (ECCV)*, 2022. 1, 3

[60] Hao Tan, Jie Lei, Thomas Wolf, and Mohit Bansal. Vimpac: Video pre-training via masked token prediction and contrastive learning. *arXiv preprint arXiv:2106.11250*, 2021. 2, 4

[61] Ziyu Wan, Jingbo Zhang, Dongdong Chen, and Jing Liao. High-fidelity pluralistic image completion with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4692–4701, 2021. 2

[62] Lin Wang, Yujeong Chae, Sung-Hoon Yoon, Tae-Kyun Kim, and Kuk-Jin Yoon. Evdistill: Asynchronous events to end-task learning via bidirectional reconstruction-guided cross-modal knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 608–619, 2021. 1, 3

[63] Rui Wang, Dongdong Chen, Zuxuan Wu, Yinpeng Chen, Xiyang Dai, Mengchen Liu, Yu-Gang Jiang, Luowei Zhou, and Lu Yuan. Bevt: Bert pretraining of video transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14733–14743, 2022. 2

[64] Zuowen Wang, Yuhuang Hu, and Shih-Chii Liu. Exploiting spatial sparsity for event cameras with visual transformers. *arXiv preprint arXiv:2202.05054*, 2022. 6

[65] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9653–9663, 2022. 2

[66] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*, 2022. 6

[67] Xumin Yu, Lulu Tang, Yongming Rao, Tiejun Huang, Jie Zhou, and Jiwen Lu. Point-bert: Pre-training 3d point cloud transformers with masked point modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19313–19322, 2022. 2, 4

[68] Alessandro Zanardi, Andreas Aumiller, Julian Zilly, Andrea Censi, and Emilio Frazzoli. Cross-modal learning filters for rgb-neuromorphic wormhole learning. *Robotics: Science and System XV*, page P45, 2019. 1, 3

[69] Alessandro Zanardi, Julian Zilly, Andreas Aumiller, Andrea Censi, and Emilio Frazzoli. Wormhole learning. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 7899–7905. IEEE, 2019. 3

[70] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12104–12113, 2022. 6

[71] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. *arXiv preprint arXiv:2111.07832*, 2021. 2

[72] Alex Zihao Zhu, Ziyun Wang, Kaung Khant, and Kostas Daniilidis. Eventgan: Leveraging large scale image datasets for event cameras. In *2021 IEEE International Conference on Computational Photography (ICCP)*, pages 1–11. IEEE, 2021. 1, 3

[73] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. Ev-flownet: Self-supervised optical flow estimation for event-based cameras. *arXiv preprint arXiv:1802.06898*, 2018. 1, 3

7. Convergence Curve

In Fig. 6 we show that finetuning convergence is much quicker for MEM than ViT-from-scratch. This saves significant computational power during finetuning to a specific classification task. Additionally, the final accuracy value is higher than ViT-from-scratch (+18.66% gain on N-Caltech101 [44]).

8. Token and Masked Prediction Visualizations

We visualize additional codebook vectors in Fig. 7. Notice how each codebook index corresponds to a specific visual feature. The most common codebook vectors are completely blank since the event histogram is sparsely populated with event count values. Due to redundancy, we do not visualize these blank codebook vectors. Although all codebook vectors are fixed, the decoder adapts each patch to its surroundings to form a coherent image. Hence, the visualized examples of a decoded codebook vector show slightly different appearances (notice the visual variations along the columns of Fig. 7). All visualized codebook vectors are rendered from the N-Cars [55] test set. We also visualize additional masked patch reconstructions during pretraining in Fig. 8 for all datasets used in this work.

9. Implementation Details

9.1. ViT Architecture

We use the ViT-B16 architecture described in [16], using 12 layers, 12 heads, an embedding size of 768, and an MLP size of 3072. The patch size is 16×16. During pretraining, we use a temporary linear layer (MEM layer) at the output of the ViT, which predicts the visual tokens. During finetuning, we replace the MEM layer with a linear layer for classification. We employ relative positional encoding [16].

9.2. Hyperparameters

We report the hyperparameters for training the dVAE in Tab. 5, for pretraining in Tab. 6, and for finetuning in Tab. 7. We use the official train and test splits for N-ImageNet [32] and N-Cars [55]. For N-Caltech101 [44], we randomly split the data into 80% training data and 20% test data.

9.3. Details on Event Preprocessing

After loading all events of a sample, we slice the events in time by randomly selecting one contiguous batch of up to 30,000 events. During training, we perform (i) a random time flip, which amounts to reversing the polarity ($p = 0.5$), (ii) a random horizontal flip ($p = 0.5$), (iii) a random shift of x-coordinates by Δx and y-coordinates Δy, where we sample $\Delta x \sim \mathcal{U}(-15, 15)$ and $\Delta y \sim \mathcal{U}(-15, 15)$.

We accumulate the augmented events into a two-channel histogram. For N-Caltech101 and N-Cars, we resize the histograms to 224×224. For N-ImageNet, we first resize the histogram to 256×341 and subsequently randomly crop the image to 224×224. Next, we remove hotpixels, a type of noise specific for event cameras, which manifests as continuously triggering events [28]. We define a pixel as a hotpixel if its event count is ten standard deviations above the mean value in the event batch. We normalize the histogram to $(0, 1)$. Lastly, during training, we perform RandAugment [11] with magnitude 20 and 2 operations. All three stages of MEM (dVAE, pretraining, and finetuning) share the same preprocessing.

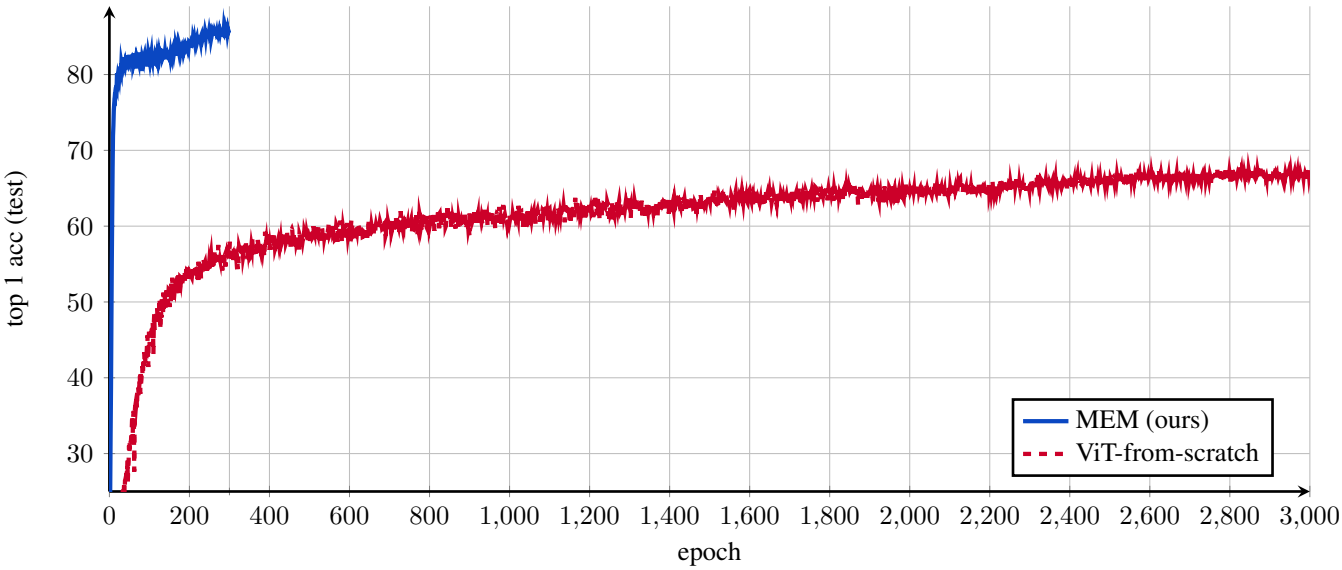

Figure 6. Finetuning accuracy vs. epochs on N-Caltech101 [55]. With our proposed pretraining (MEM), the accuracy increases much faster. It reaches a higher final accuracy of 85.60% compared to finetuning without pretraining (ViT-from-scratch), where the final accuracy is only 66.94%. Both the pretraining and the finetuning tasks use the entire N-Caltech101 train dataset.

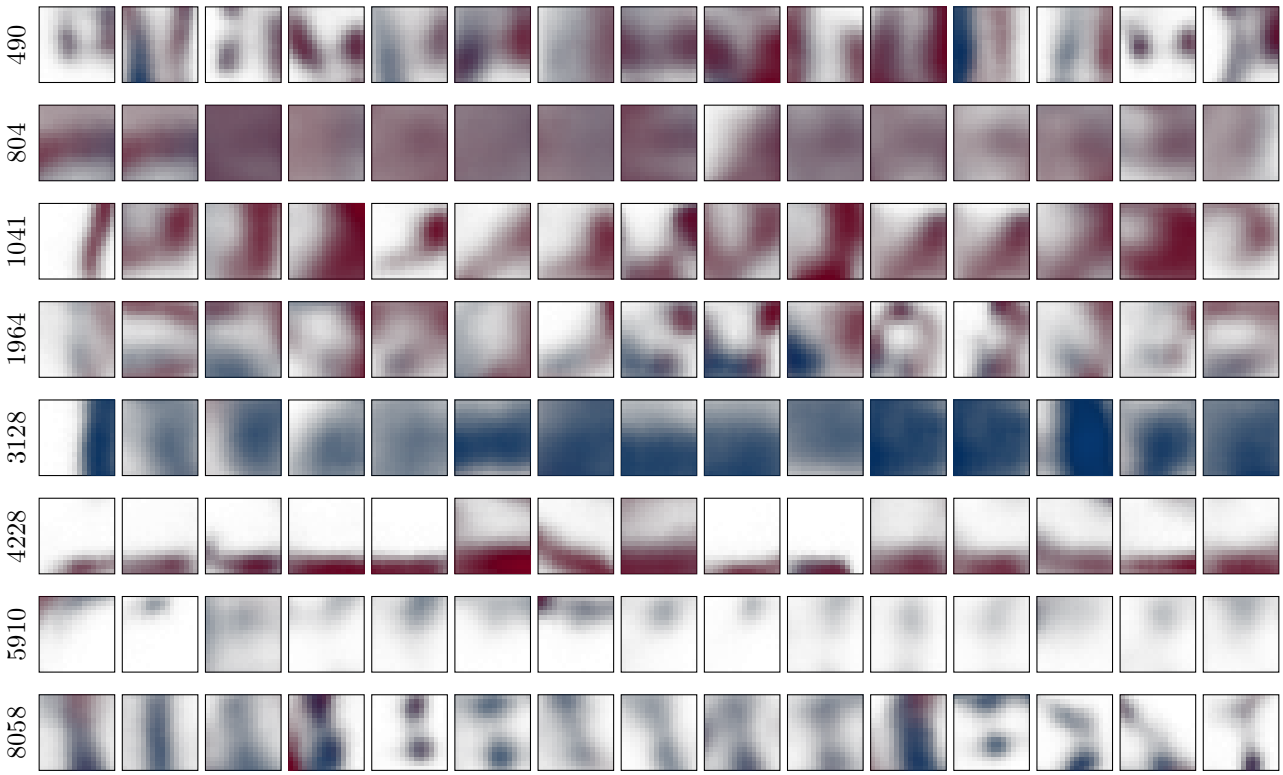

Figure 7. Additional examples of decoded codebook vectors with the codebook indexes of 490, 804, 1041, 1964, 3128, 4428, 5910, and 8058. Although all codebook vectors are fixed, the decoder adapts each patch to its surroundings to form a coherent image. Notice how each codebook index corresponds to a specific visual feature, e.g., a red horizontal line at the bottom of the patch (4228) or a round mixture of red and blue polarity (1964). The codebook size is 8092. These visualizations are rendered from the test set of N-Cars [55].

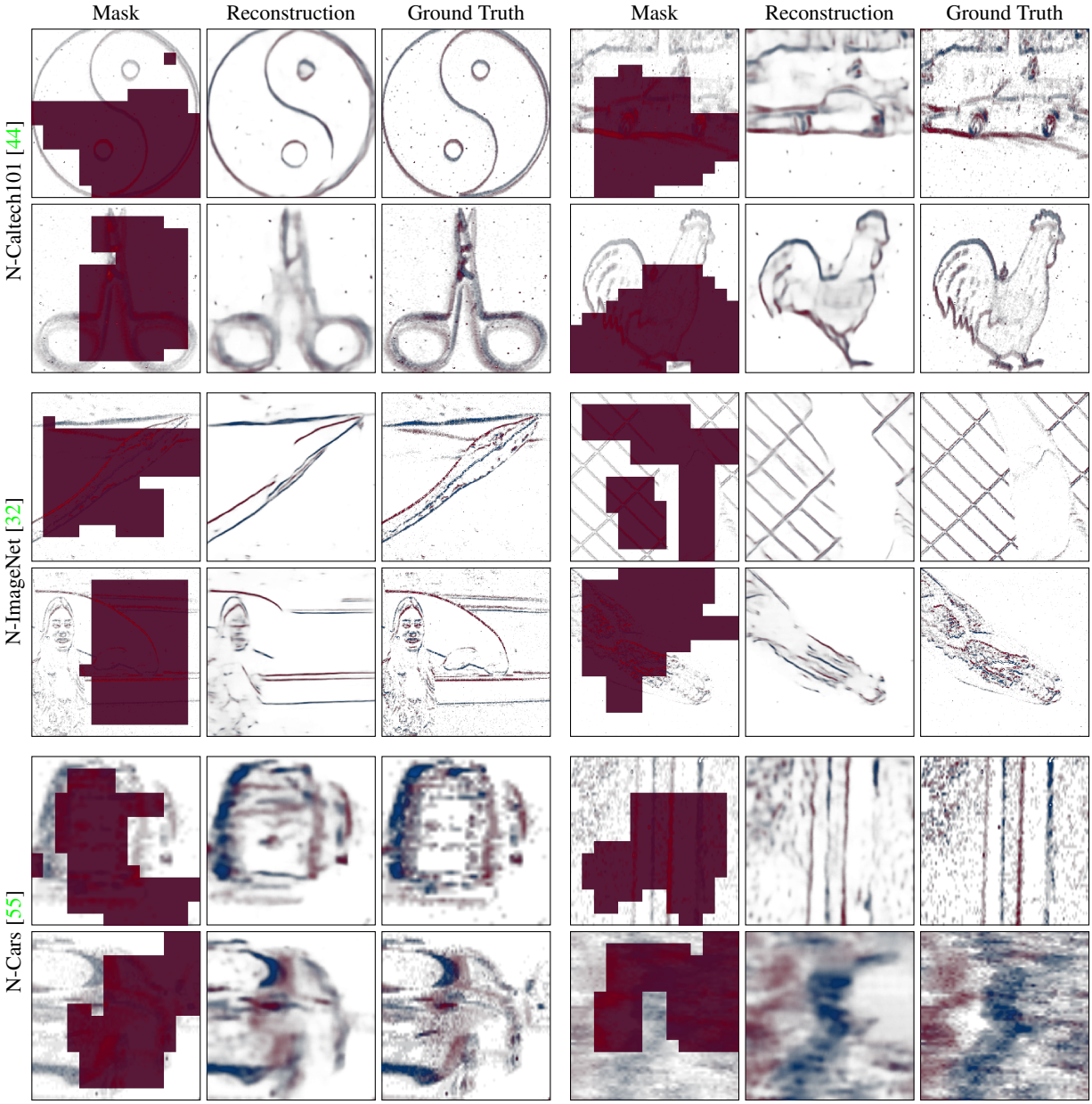

Figure 8. Additional masked patch predictions. From left to right: We visualize the masked input histograms, the reconstructions during pretraining, and the ground truth for N-Caltech101 [44] (top) and for N-Imagenet [32] (middle) and for N-Cars [55] (bottom). Note how the ground truth can be recovered even if large parts of the input to the ViT are masked. The ViT predicts the tokens for all masked patches, which are then decoded into the predicted event histogram by the decoder of the dVAE. These visualizations are rendered from the test set.

Hyperparameter	N-ImageNet [32]	N-Caltech101 [44]	N-Cars [55]
optimizer	Adam [33]	Adam [33]	Adam [33]
optimizer momentum	$\beta_1, \beta_2 = (0.9, 0.999)$	$\beta_1, \beta_2 = (0.9, 0.999)$	$\beta_1, \beta_2 = (0.9, 0.999)$
learning rate	1e-3	2e-4	2e-4
learning rate schedule	exponential (0.99)	exponential (0.99)	exponential (0.99)
learning rate layer decay	0.98	0.98	0.98
kl weight	1e-10	1e-10	1e-10
batch size	512	192	192
grad clip	1e-2	1e-2	1e-2
epochs	50	300	300

Table 5. Hyperparameters for the dVAE.

Hyperparameter	N-ImageNet [32]	N-Caltech101 [44]	N-Cars [55]
optimizer	AdamW [38]	AdamW [38]	AdamW [38]
optimizer momentum	$\beta_1, \beta_2 = (0.9, 0.95)$	$\beta_1, \beta_2 = (0.9, 0.95)$	$\beta_1, \beta_2 = (0.9, 0.95)$
learning rate	1e-4	5e-4	3e-4
learning rate schedule	cosine decay [37]	cosine decay [37]	cosine decay [37]
warmup steps	1000	1000	1000
weight decay	0.05	0.05	0.05
batch size	512	512	384
grad clip	30	30	30
epochs	75 †	3000	1000‡

Table 6. Hyperparameters for pretraining. †Cosine scheduler set for 300 epochs, but for computational reasons, only training for 75 epochs. ‡ Cosine scheduler set for 3000 epochs, but only training for 1000 epochs.

Hyperparameter	N-ImageNet [32]	N-Caltech101 [44]	N-Cars [55]
optimizer	AdamW [38]	AdamW [38]	AdamW [38]
optimizer momentum	$\beta_1, \beta_2 = (0.9, 0.95)$	$\beta_1, \beta_2 = (0.9, 0.95)$	$\beta_1, \beta_2 = (0.9, 0.95)$
learning rate	1e-3	4e-3	5e-4
learning rate schedule	cosine decay [37]	cosine decay [37]	cosine decay [37]
learning rate layer decay	0.65	0.65	0.65
warmup epochs	20	20	20
weight decay	0.3	0.05	0.05
drop path	0.1	0.1	0.1
dropout	0.0	0.1	0.1
batch size	1024	1024	1024
grad clip	30	30	30
epochs	200†	300	300

Table 7. Hyperparameters for finetuning. †Cosine scheduler set for 300 epochs, but for computational reasons, only finetuning for 200 epochs. We report the exponential moving average accuracy on N-Imagenet with a decay factor of 0.9999.